Toward classifying randomly asynchronous cellular automata

N. T. Clapham, M. (Spike) Barlow and R. I. McKay

School of Computer Science, University College,
University of New South Wales, Australia
Email: {nclapham,spike,rim}@cs.adfa.edu.au

Abstract.
In some respects we can classify cellular automata into the dynamical categories of ordered, complex and chaotic, by the degree of their order or regularity. However, in the case of randomly asynchronous automata it is not obvious whether a classification can be made into the traditional categories as their synchronous counterparts. Can order emerge in an asynchronous environment, where irregular updating produces an indeterministic path through the state-space? In this paper we employ 1 auto-correlation and 3 entropy variance functions as measures of complexity. We find that these measures provide a method to classify randomly asynchronous cellular automata into 4 classes. Despite all classes showing varying levels of dynamical instability, we can identify a level of order inherent in the first 2 classes. We suggest that these classes sustain their order by maintaining a few stable structures, allowing others areas more dynamical freedom.

1. Introduction

"Order, order" the speaker demands in the midst of a chaotic parliamentary debate; the room subsides before erupting again into another debate. In some respects the degree of order, or regularity, is an indication of a system’s dynamics; Immanuel Kant once said that “nature itself, even chaos, cannot proceed except in an orderly and regular manner.” What of randomness then, when order, it seems, is a property even chaos demands? Can an asynchronous system, where agents are randomly updated, be classified into an ordered, complex or chaotic domain? Finding order in a parliamentary debate may be difficult, yet ordered processes exist in many forms around us. It is this underlying order that we aim to understand.

In this account we investigate the dynamical properties of randomly updated asynchronous cellular automata. We address the question of whether the dynamics of these automata can be classified into the categories of ordered, complex and chaotic? Four classificatory measures are identified and used to classify automata. Our conjecture is that randomly asynchronous cellular automata can be classified into these dynamical categories just as their synchronous counterparts. In the introductory sections we will introduce cellular automata and discuss the properties of asynchronous updating. Following this discussion we define four measures to classify cellular automata, describe experiments and conclude with a discussion of the results.

2. Cellular automata

Cellular automata (CA) are composed of a network of automatons. As contrast to the general graph of a Boolean Network (BN) (Kaufman 1992), CA are restricted to preserve spatial locality. That is, CA models are commonly defined as a 1D lattice, 2D landscape or 3D solid – their connectivity is ordered and regular. Conway’s game of life (Conway 1982) is an example of a 2DCA. The usefulness of CA come from the ability to observe and analyse emergent behaviour arising from a locally connected system.

Formally, we can define cellular automata as identically programmed automatons that locally interact with one another. CA can be viewed as a spatial array of N cells C in some dimension D (eg. figure 1). Each cell is situated within a local neighbourhood η that includes itself. At any point in time a cell is in one of a finite number of states. For our purposes we will restrict our discussion to boolean systems (ie. two states: 0 or 1). Evolution of a cell’s state C^t over time t occurs as a function of the interaction between its local neighbourhood: C^t+1 = f(η). In this paper we will limit our discussion to 1DCA with a neighbourhood radius of 2, therefore a cell’s neighbourhood includes itself along with the two adjacent cells to the left and right: |η| = 5.

Each automaton within the system contains an identical rule-set. The rule-set for a two-state automaton with a neighbourhood η of 5 contains 2^5 rules. Thus, rule-tables can be represented by a
32-bit binary number, where the output for a neighbourhood input 00000 corresponds to the bit in position 0 of the binary array. There are therefore a total of $2^3$ possible rule-tables, a significantly large number.

Typically, CA are updated synchronously. However, in the context of many real-world complex systems, strictly synchronous behaviour is hard to justify (Somogyi & Snegoski 1996)[Ingerson & Buvel 1984]. In this paper we make the assumption of random asynchrony and analyse the dynamics arising from the model. The method of updating the system is as follows: a cell is chosen, with replacement, over a uniformly random distribution, for update; after $N$ randomly chosen cells have been updated the global system state is recorded. This process is then repeated for $T$ time steps.

A common method to view the behaviour of a 1DCA is a space-time plot, which illustrates the evolution of the system’s state through time. At every time step $T$ the current state of each cell is displayed on a lattice. Cells in state 0 are black and those in state 1 are white. Time proceeds from top to bottom. This plot provides an effective method to view patterns formed by the automaton rule set. See examples of space-time plots in figures 4, 5, 6 and 7.

3. Properties of randomly asynchronous update

Before we begin to describe some properties of random asynchrony we will define some useful concepts. The first of these concepts is the idea of an initial-condition ($IC$). An $IC$ is the starting state of the system, from which evolution occurs. This initial system state is defined as the set of all cell states $C_s$ within the system at time zero. Thus,

$$IC \in \{C_1, C_2, ..., C_N\}_{t=0}$$

Once the system is initialised to an $IC$ the system evolves from one state to the next. Tracing these state changes throughout time provides a state transition diagram or a picture of the system trajectory from the $IC$. This state transition diagram forms a directed graph. If we imagine the trajectories from the set of all $IC$, there are $2^N$ of these, we have a graph of the system state or phase-space.

Visualising the set of all trajectories, or the state-space, allows us to view a system’s dynamics in the terms of attractors. Attractors are a method of describing the long-term dynamical behaviour of a system. Traditionally, attractors are broken into three categories: point attractors, periodic attractors and strange attractors. Point attractors lead to a steady unchanging system state; periodic attractors cycle through a set of states; and strange attractors describe chaotic dynamics.

Viewing the dynamics of a system in the terms of attractors leads to another observation; in a synchronous system, each $IC$ leads to a particular attractor. That is, every system state lies on a path along the state transition graph that leads to a particular attractor. Therefore, every attractor consumes a portion of the state-space, called an attractor’s basin.

We can now contrast some general properties of randomly asynchronous and synchronous updating:

1. indeterminism;
   Synchronous updating produces a deterministic path through the state-space. That is, the system state $a_t$ is directly determined by state $a_{t-1}$ because $a_{t-1}$ always produces state $a_t$. However, rather than producing a deterministic evolution, the randomly asynchronous model does not deterministically produce state $a_t$ from state $a_{t-1}$. This indeterminacy is introduced by the randomness of the update function. That is, the order in which cells are updated is crucial to the trajectory through the state-space.

2. attractor basins are not discrete;
   In the deterministic case attractor basins are discrete, each state leading directly to its attractor. However, as there is a probability attached to the transition between system states in the randomly asynchronous case, system states do not discretely lie within any one attractor’s basin. Therefore, the basins of a system’s attractors have indistinct boundaries, in that one state could lead to one of two or more possible attractors.

3. no ‘strictly’ periodic attractors;
   As synchronous updating produces a deterministic pathway through the system state-space, periodic attractors always cycle through a fixed set of states. This cycle leads to regular static
patterns in the systems dynamics. However, in the randomly asynchronous case state transitions are not deterministic. Consequently, strictly periodic attractors, as observed in the synchronous case, cannot exist.

To deal with these differences Harvey and Bosmaier (Harvey & Bosmaier 1999) define the loose attractor. A loose attractor is an attractor that passes indefinitely through a sub-set of all possible states. For example, if \( R \) is the set of all reachable states (ie, those that form the attractors), then \( A \) is a loose attractor if \( R \in A \) and for all possible successor states, \( R' \in A \). The difference from a strictly periodic attractor is that the path through the sub-set of states is not fixed.

Considering the above differences it is not obvious that a randomly asynchronous CA could be characterised by the traditional dynamical properties of its synchronous counterpart. There are no ‘strictly’ cyclic attractors, attractor basins are not discrete and the state-space trajectories are not deterministic. Is it possible to find parallels to the synchronous dynamics?

4. Classifying cellular automata

Synchronously updated cellular automata are typically classified into 4 classes, as defined by Wolfram (Wolfram 1984). Class 1 evolves to a homogeneous state, analogous to a limit point attractor; class 2 leads to simple and stable periodic structures, analogous to limit cycles; class 3 produces a chaotic pattern, analogous to strange attractors; and class 4 displays complex localised structures, sometimes long-lived.

Langton (Langton 1990) suggests that class 4 lies at a phase transition between the order and chaos of classes 2 and 3. Further, Wensche (Wensche 1999) suggests the combination of classes 1 and 2 – as many ordered rules exhibit both limit point and limit cycle attractors. For these reasons Wensche (Wensche 1999) adjusts the classification as follows: ordered (classes 1 and 2) – complex (class 4) – chaotic (class 3).

Another method for classifying CA dynamics is the \( \lambda \) parameter (Langton 1990). The \( \lambda \) parameter simply measures the density of resting states produced by the rule table. Therefore, in a 2 state CA, \( \lambda = s_q / 2^{11} \), where \( 11 \) is the size of the neighbourhood and \( s_q \) is the count of 1’s (the arbitrarily chosen resting state) in the rule table. As \( \lambda \) increases, a phase transition between periodic and chaotic behaviour occurs, with the most ‘complex’ behaviour observed on the transition boundary.

5. Entropy variance as a measure of complexity

One way to identify the dynamical behaviour of a system is to measure the ratio of order to disorder within a system’s individual components. When the ratio of order is high the system is more likely to tend toward orderly dynamics. Likewise, when disorder dominates the opposite is true; the system tends toward disorderly dynamics. Perhaps complexity can be defined when this ratio peaks, where order acts as a glue holding the system together, not allowing disorder to escape. However, rather than claiming this ratio to be a general measure of complexity, we employ this measure as an indication of the order inherent within the cellular automata.

Consider the Shannon entropy function:

\[
H(S) = - \sum_{k=1}^{n} p_k \log p_k
\]

Where \( p_k \) is the probability of the event \( k \) and \( H(S) \) is the entropy of system \( S \). When the probability \( p_k \) is 0 the value 0 is assigned to \( p_k \log p_k \). The entropy is defined as the average of the self-information, where self-information is defined as \( -\log p_k \). This measure increases as uncertainty about events grow, when there is complete certainty \( H(S) = 0 \), the entropy vanishes. One point to make here is that as randomness increases so does the entropy, which is exactly what the measure is supposed to identify. However, we are not interested in a system with high entropy, as such a system displays no ordered properties. Moreover, we require a measure to capture the relationship between order and disorder. Therefore we turn to the variance of the entropy of each cell within the CA.

When the entropy variance is 0, either complete order or complete disorder exists, as each agent has a low entropy or a high entropy. An increase in variance indicates that a higher mixture of ordered and disordered cells are evident. The variance of the entropy of each cell is a simple measure of the relationship between order and disorder within the system.

In the following section we describe four measures that capture different dynamical properties of the CA. The first measure is auto-correlation and is a measure of correlated states through time. This measure captures quasi-periodic behaviour. The next three: rule-table complexity, space-time complexity and input-entropy complexity; are all entropy variance measures described below that identify ordered and disordered properties. The correlation

5.1. Auto-correlation

A method that gives an indication of the system’s global dynamics is a correlation function as used in (Di Paolo 2000). This measure specifically detects quasi-periodic behaviour, in contrast to the entropy-variance measures described below that identify ordered and disordered properties. The correlation
measure is defined as the average similarity of cell states \( C_i s \) at two different time steps \( t \) and \( t' \), thus

\[
Corr(t, t') = \frac{1}{N} \sum_{i=1}^{N} C_i s(t) C_i s(t')
\]

(3)

where \( s(t) \) and \( s(t') \) is a linear mapping between 1 and -1. This linear mapping simply allows a count of identical states when states are multiplied together at \( t \) and \( t' \). The auto-correlation coefficient is then defined as

\[
AC(k) = \frac{1}{M} \sum_{i=1}^{M} Corr(t, t + k)
\]

(4)

where \( k \) is the ‘lag’ or time ahead to perform the autocorrelation averaged over \( M \) steps. In the following experiments the autocorrelation was calculated for 160 time steps, \( k = [1 : 160] \), and averaged over 9000 time steps, \( M = 9000 \).

5.2. Rule-table complexity

Rule-table complexity is defined as the variance of the entropy within each cell’s rule-table over time \( T \). Whereas the input-entropy complexity described below takes a global measure of the systems rule-table entropy at each time step; rule-table complexity is a local entropy measure of each automaton’s rule-table. The rule-table entropy (RTE) of cell \( i \) in the system \( S \) is defined as

\[
RTE(S_i) = \sum_{r=1}^{2^{|l|}} \frac{F^i_r}{T} \log \frac{F^i_r}{T}
\]

(5)

where \( F^i_r \) is frequency of ‘hits’ upon rule \( r \) of cell \( i \) in system \( S \). The rule-table entropy is calculated over \( T \) time steps as a function of the probability distribution of rule-table ‘hits’. Rule-table complexity (RTC) is then defined as the variance of \( RTE(S) \), where \( n \) is the number of cells and \( n - 1 \) calculates the unbiased variance.

\[
RTC(S) = \sum_{i=1}^{n} \frac{(RTE(S_i) - \bar{RTE}(S_i))^2}{n - 1}
\]

(6)

5.3. Space-time complexity

Space-time complexity (STC) is defined as the variance of the entropy of each cell’s state over time. The space-time entropy (STE) of cell \( i \) in the system \( S \) is defined as

\[
STE(S_i) = \sum_{s=1}^{k} \frac{F^i_s}{T} \log \frac{F^i_s}{T}
\]

(7)

where \( k \) is the number of states and \( F^i_s \) is the frequency of the state \( s \) occurring in cell \( i \). The space-time entropy of cell \( i \) is calculated over \( T^* \) time steps as a function of the probability distribution of state occurrences. Space-time complexity is then defined as the variance of \( STE(S) \), where \( n \) is the number of cells and \( n - 1 \) calculates the unbiased variance.

\[
STC(S) = \sum_{i=1}^{n} \frac{(STE(S_i) - \bar{STE}(S_i))^2}{n - 1}
\]

(8)

5.4. Input-entropy complexity

A measure of this kind was used in Wunsch (Wunsch 1999) to categorise synchronously updated cellular automata into the dynamical categories of ordered, complex and chaotic. Input-entropy complexity (IEC) is defined as the variance of the entropy of the rule-lookup-frequency over time \( T \). The input-entropy (IE) at time \( t \) of system \( S \) is defined as

\[
IE(S_i) = \sum_{r=1}^{R} \frac{F^i_r}{R} \log \frac{F^i_r}{R}
\]

(9)

where \( F^i_r \) is the frequency-lookup of rule \( r \) at time \( t \) and \( R \) is the number of rules. This measure is an input-entropy measure, as used in Wunsch (Wunsch 1999), in the sense that the neighbourhood input configurations to each cell define the rule to be applied. The input-entropy complexity (IEC) is then defined as the variance of \( IE(S) \) over time \( T \), where \( T - 1 \) calculates the unbiased variance.

\[
IEC(S) = \frac{\sum_{t=1}^{T} (IE(S_t) - \bar{IE}(S_t))^2}{T - 1}
\]

(10)
6. Experiments

To test the above measures samples were taken from the entire range of possible rules for a CA with a neighbourhood of 3; that is, a cell has two neighbours to each side of itself. The total size of the rule-table is $2^3$, with a total of $2^{25}$ or $2^{32}$ possible rules. If we order the number of possible rules according to the number of transitions to $1$ in the rule-table, that is 0 to 32 transitions, we have a binomial distribution. Therefore samples were taken spread over this same distribution, where most samples were taken from the centre of the distribution, where the bulk of the rules lie. Approximately 3000 samples were taken in all.

The size of the CA was $N = 128$ cells wide and the total time steps for the evolution of each run was $T = 10000$. Each time-step was broken into 128 random updates; that is, every 128 uniformly random updates, with replacement, the system state was recorded and time $T$ was advanced one time step. Each of the measures described above: $AC$, $RTC$, $STC$, and $IEC$ were applied to every sample. All the measures were normalised into a range of 0 to 1.

Figure 2 shows the distribution of data points for each measure. Figure 2(a) is a plot of all samples sorted from min to max for each measure, which illustrates the small number of samples receiving high values in each case. Figure 2(b) is the histogram of each distribution. In all cases only relatively few automata received high values for the given measures.

For comparison with the feature value, figure 3 illustrates the relative distribution of each measure. This graph shows that in all cases high values range within $\lambda = 0.2$ to $\lambda = 0.8$. One point to note here is that the dynamical activity described by $\lambda$ is only loosely correlated in a system with a small number of cell states.

For each of the four measures space-time plots of the six highest CA are illustrated in figures 4, 5, 6 and 7. Each of the measures seems to identify a different dynamical behaviour, four in all. In this account, rather than attempting to identify these classes as analogues of the Wolfram (Wolfram 1984) four classes, these four classes of dynamical behaviour will be referred to as class A, B, C and D. In the following sections, we discuss the results for each of these four classes.

6.1. Class A; auto-correlation

Class A automata are identified by high variance in the autocorrelation measure. This measure indicates that the global dynamics display quasi-periodic structures in the space-time evolution. Figure 4, from left to right, illustrates the 6 CA that received the highest values measured with the autocorrelation function.

All samples taken that had a normalised variance, from 0 to 1, of greater than $\approx 0.5$ did not end their evolution as point attractors. Runs ending as point attractors started appearing when the variance was less than $\approx 0.5$. However, examining runs around this point did not indicate a transition from quasi-periodic to ‘less quasi-periodic’ structures.

The high variances in class A seem to distinguish CA that typically display dislocations of an underlying pole structure that travels around the lattice. This traveling wave shows a small degree
of dynamical instability, however, the macro-structure is held together and still exhibits elements of order.

6.2. **Class B; rule-table complexity**

Class B automata are identified by a high rule-table complexity. This measure indicates the degree to which a cell's movement within its rule-table is stable and the variance represents the relationship between ordered and disordered rule-tables. For example, in high variance a few cells have a very stable rule-table, while others are more erratic in the movement around their rule-table. Figure 5, from left to right, illustrates the 6 CA that received the highest values measured with the RTC function.

Point attractors start to appear as the normalised variance dropped below $\approx 0.25$. Again, examining the automata around this point does not indicate a transition where the measure stops identifying the characteristics shown in figure 5.

The rule-table complexity measure seems to distinguish automata that contain stable pole structures that hold together a small degree of dynamical instability. In some cases the poles act as a barrier, preventing other unstable activity from escaping; while in other cases the poles act as a dividing range, separating areas of instability. Overall, the system continues showing some order despite areas of instability.

6.3. **Class C; space-time complexity**

Class C automata are identified by a high space-time complexity. This function captures the degree to which order exists over time in cell states. Figure 6, from left to right, illustrates the 6 CA that received the highest values measured with the STC function.

Point attractors started appearing when the normalised variance dropped below $\approx 0.5$. Yet again, examining this point does not reveal a transition in the measure's identification of automata. Another observation is that the STC measure is the only measure to show any correlation with another measure; at low variances the STC is slightly correlated with the rule-table complexity measure. Therefore at lower variances ($\approx 0.25$) RTC and STC identify similar dynamical characteristics.

The dynamical behaviour of this class is difficult to describe, as no strong sense of order exists. However, high variances are loosely characterised by small irregular ‘disasters’ that clear a section of cells. The first, second and fourth show some rotational characteristics whereas the third, fifth and sixth do not. At lower variances the dynamics show characteristics similar to class B.

6.4. **Class D; input-entropy complexity**

Class D automata are identified by a high input-entropy complexity. This measure indicates the degree to which order exists in a cell's neighbourhood input or the global rule-table. Figure 7, from left to right, illustrates the 6 CA that received the highest values measured with the IEC function.

Point attractors in this class behave differently. Rather than increasing in number as the measure falls, runs ending in point attractors appear only when the measure is greater than $\approx 0.35$. It is typical for a run above this point to become a point attractor after $\approx 5000$ evolutions. Point attractors could perhaps boost the variance as there is a sharp contrast in the entropy over time when every cell remains in a stable state after some instability for $\approx 5000$ evolutions. The dynamics of class D can be characterised by a fractal dimension, with complex structures propagating through time.
Figure 4: Class A randomly asynchronous cellular automata. From left to right, the 6 CA that received the highest values measured with the auto-correlation function. The first 500 time-steps are shown. This class is characterised by short quasi-periodic cycles traveling around the torus.

Figure 5: Class B randomly asynchronous cellular automata. From left to right, the 6 CA that received the highest values measured with the rule-table complexity. The first 500 time-steps are shown. This class is characterised by stable pole structures that hold together a small amount of dynamical instability.
7. Discussion

It seems that high variances using the measures presented here reveal 4 different classes of randomly asynchronous cellular automata; classes A, B, C and D. There is one immediately obvious point, all classes show varying levels of dynamical instability, however, in classes A and B this instability is not so great as to be called chaotic. In classes A and B a high level of order is evident, even though random asynchrony is employed. For this reason, we could suggest that classes A and B belong to the “ordered” category. Class D tends toward more complex dynamics while class C tends toward a more disordered region. However, these categorisations tend to be somewhat subjective at this point.

Why, in classes A and B is such order evident? Perhaps this order can be attributed to the high stability and order of a few individual cells. In class A, it seems, every cell has a level of order, where cells return to a stable state after they have been disturbed. It seems that the underlying pole structure is stable, with dislocations, or disturbances, of the poles traveling around the lattice.

In the case of class B there exists a few highly ordered cells that seem to act as a cohesive component in the overall dynamics. The poles traveling down the system seem to constrain the dynamical instability of the other cells, not allowing their disorder to escape. Perhaps the interesting question is why the areas of instability do not themselves eventually turn into stable poles?

In the case of class A and B the high stability of a few components allows a level of ‘order’ to be sustained throughout the system’s evolution. Another property that may play a part in these ‘stable structures’ is the underlying 1 dimensional lattice. In some sense this 1-dimensional condition places a restriction upon the system which could play a role in the emergence of the ordered components.

8. Conclusion

We have come some way to addressing the conjecture that randomly asynchronous cellular automata can be classified into the dynamical categories of ordered, complex and chaotic. We have identified four separate classes of randomly asynchronous cellular automata using four different measures, with each class exhibiting varying degrees of order.

Class A displays quasi-periodic space-time patterns, typically dislocations of an underlying pole structure traveling around the lattice; class B is typically characterised by stable pole structures that hold together a small amount of dynamical instability; class C is difficult to describe as no strong sense of order is evident, however, it seems to be loosely characterised by small irregular ‘disasters’ that clear a section of cells; class D is characterised by a fractal dimension, with complex structures propagating through time.

Although all classes show different levels of dynamical instability, classes A and B could perhaps be described as ordered classes; class D as complex; and class C as relatively disordered. One interesting observation is that the order in classes A and B are held together by strong stable components, perhaps a factor of the underlying 1-dimensional lattice. This cohesion, that is, the stable structures
holding together a small degree of instability, may suggest that order is sustained in these systems by maintaining a few stable structures, allowing other areas more dynamical freedom.

The categories presented above are a somewhat subjective classification of our ability to recognise patterns. Therefore, further study is required to investigate the underlying properties of randomly asynchronous attractors and assert any classification with confidence.

References


